

2.6 Ereignisorientierter Modellierungsstil der diskreten Simulation

- Klassischer Modellierungsstil der Simulation
- Modellierung basiert auf Beschreibung von elementaren Systemzustandsänderungen zu diskreten Zeitpunkten
- Zusammenfassung zusammengehöriger elementarer Zustandsänderungen zu **Ereignissen**
- Detaillierte Beschreibung der einzelnen Schritte der Zustandsänderungen zum diskreten Ereigniszeitpunkt in **Ereignisprozeduren**

Ereignisorientierte Sicht der Simulation

- Zeitlich ausgedehnte Vorgänge (**Aktivitäten**) modelliert als **Folge von Ereignissen**
- Ereignisse laufen **zeitverzugslos** ab
- Interne Repräsentation des Modells ebenfalls in Form von Ereignissen
- "Bottom Up"- Modellierung
 - Abbildung: Beziehungen zwischen prozess- und ereignisorientierter Simulation
- Ereignisorientierte Simulation = Event Scheduling Simulation

Ereignisorientierte Sicht der Simulation

Softwaretechnische Realisierung eines ereignisorientierten Simulationsmodells

- **Ereignisprozeduren**
 - Beschreibung der modellspezifischen Zustandsänderungen
 - Änderung von Attributwerten von Objekten
 - Generierung/ Löschen von temporären Objekten
 - Ansetzen neuer Ereignisse
 - Verschieben/ Löschen vorgemerakter Ereignisse
- **Ereignisliste / Ereigniskalender**
 - Nach Ereigniszeitpunkten aufsteigend geordnete Liste der Ereignisse unterschiedlichen Typs

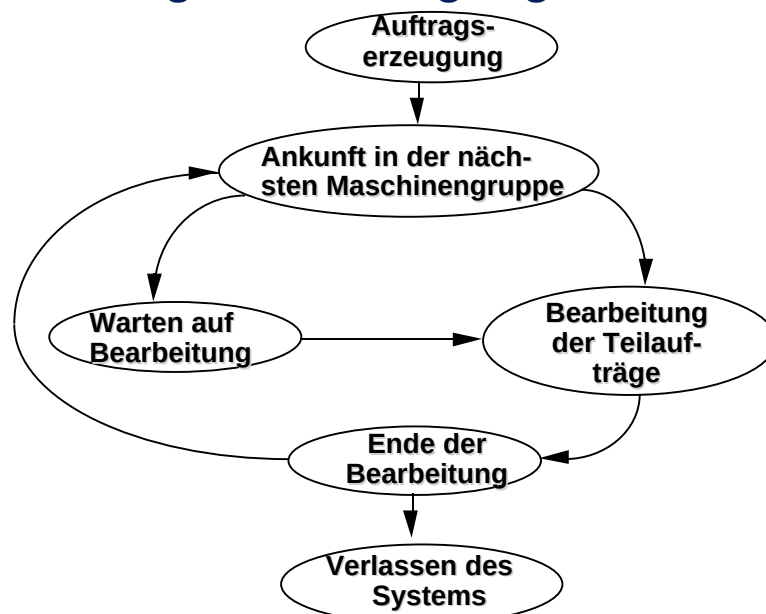
Ereignisorientierte Sicht der Simulation

Softwaretechnische Realisierung eines ereignisorientierten Simulationsmodells

- **Zeitführungsprozedur**
 - Abarbeitung der Ereignisliste nach Zeitpunkten und Aktualisierung der Simulationsuhr
- **Steuerprozedur**
 - Ereignisauswahl und Aufruf der zugehörigen Ereignisprozedur
- **Initialisierungs- sowie Auswertungsprozedur**
 - Setzen von Anfangswerten und Anfangsereignissen
 - Berechnung und Aufbereitung der Simulationsergebnisse
- **Flussdiagramm mit prinzipiellem Ablaufschema der ereignisorientierten Simulation**

Ein Simulationsbeispiel: Ereignisorientierte Version des Fertigungsmodells

- Zustandsänderungen im Fertigungsmodell**



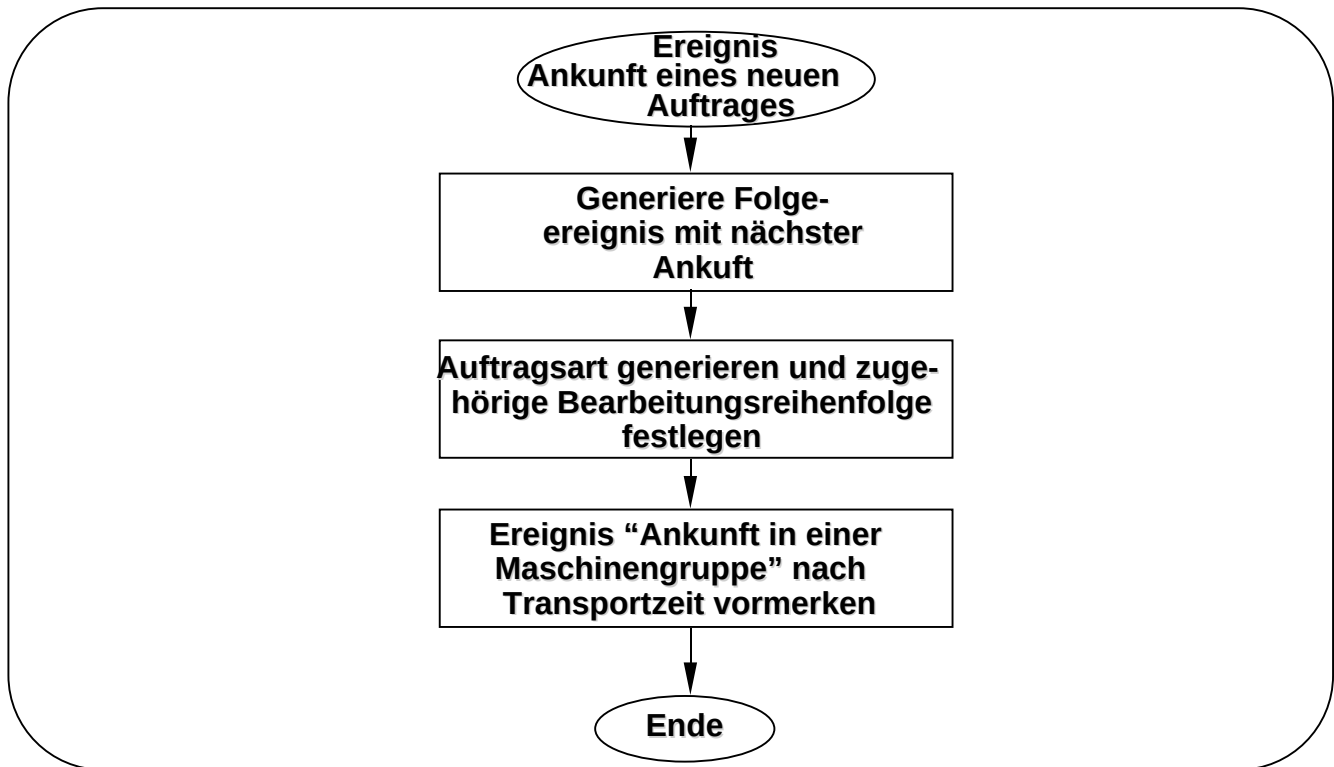
Modellaufbau des ereignisorientierten Fertigungsmodells

- Identifikation der Ereignistypen**

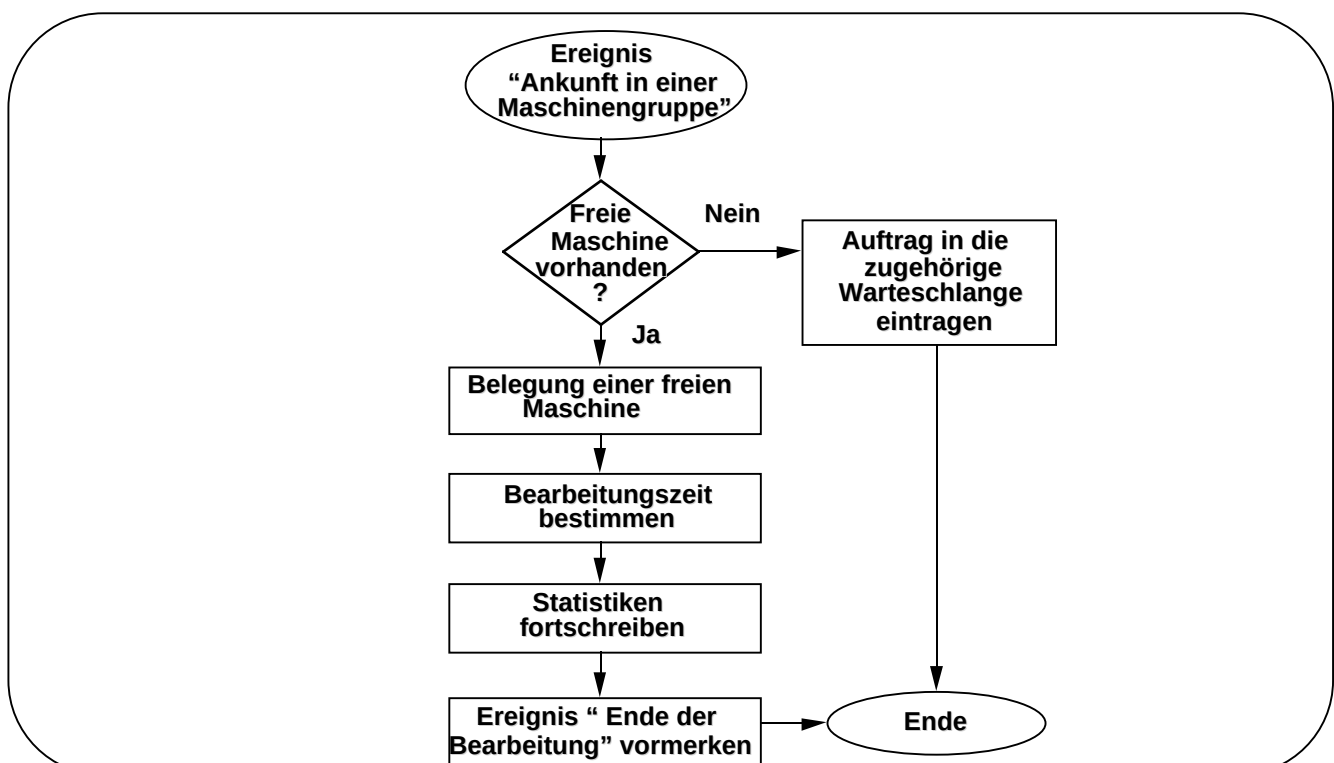
- Zusammenfassung elementarer Zustandsänderungen zu Modellereignissen

- Ereignis “Ankunft eines neuen Fertigungsauftrages im Fertigungssystem”
- Ereignis “Auftragsankunft an einer Maschinengruppe”
- Ereignis “Bearbeitungsende eines Auftrages an einer Maschine”

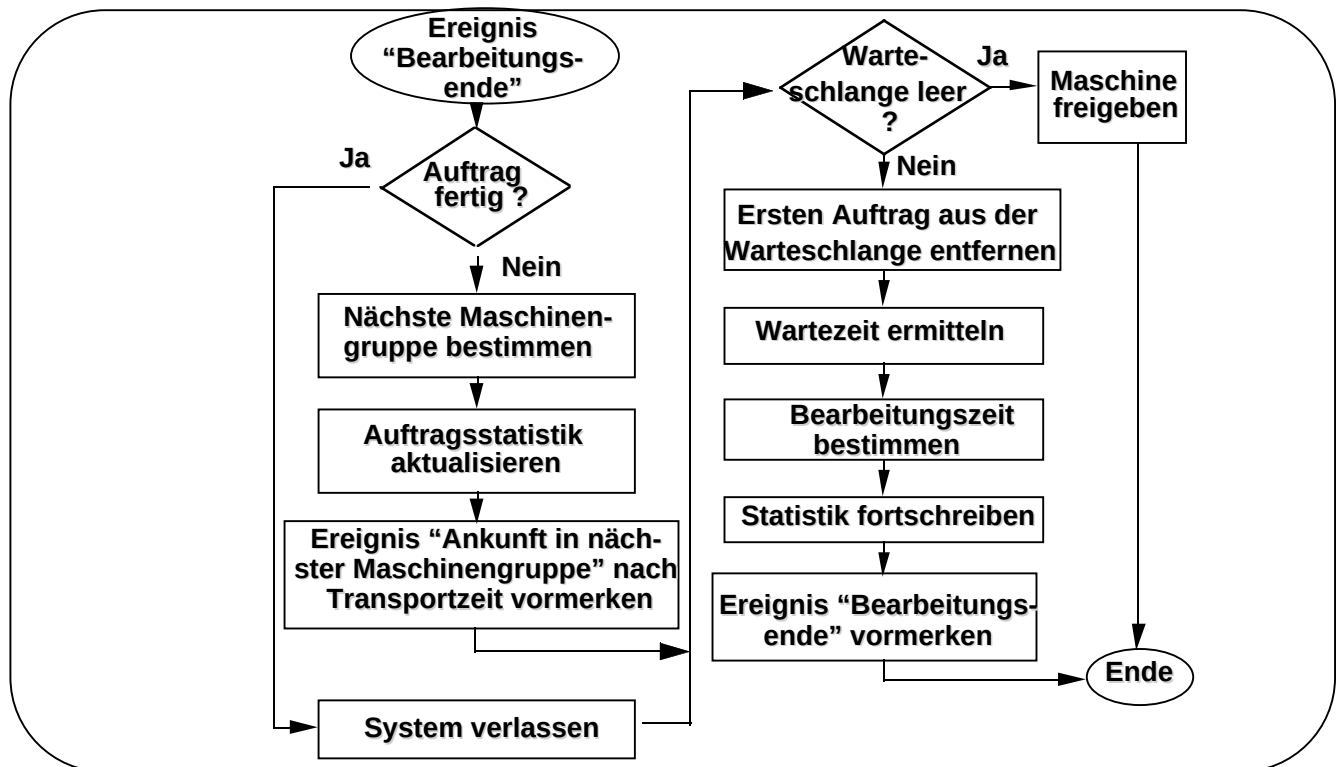
Flussdiagramm des Ereignisses "Ankunft eines neuen Fertigungsauftrages im Fertigungssystem"



Flussdiagramm des Ereignisses "Auftragsankunft an einer Maschinengruppe"



Flussdiagramm des Ereignisses "Bearbeitungsende des Auftrages"



2.7 Simulation von Lagerhaltungssystemen

- Weitere wichtige Anwendungsklasse der diskreten Simulation
- **Lager** = Puffer zum Ausgleich von Nachfrage und Produktion / Anlieferung
- **Hauptziele der Lagerhaltung**
 - Hohe Nachfragedeckung
 - Niedrige laufende Lagerhaltungskosten
 - Niedrige Kapitalbindung in Lagerbestand/Lagerinvestitionen
- **Untersuchungsgegenstand von Lagerhaltungsmodellen**
 - Kosten der Lagerhaltung
 - Erfüllbarkeit der Aufträge bei bestimmten Bestellpolitiken/Lagerkapazitäten

Simulation von Lagerhaltungssystemen

- **Unterschiede zu Bedienungs-/Wartesystemen**
 - Ressourcen (= Produkte) verbrauchbar und erneuerbar
 - Schwerpunkte auf sofortiger Erfüllbarkeit von Aufträgen, nicht auf Auftragsabfertigung (= Bedienungsvorgang)
 - Systemzustandsänderungen vor allem Bestandsänderungen
 - Ziel einer hohen Sofortlieferfähigkeit (entspricht Minimierung der Kundenwartezeiten)
- **Kombination von Lagerhaltungs- und Bedienungsmodellen möglich**

Simulation von Lagerhaltungssystemen

Systemstruktur

- **Objekte**
 - Lager = Menge von Produkten mit aktuellem Lagerbestand
 - » Bestand: Minimal-, aktueller, Sollbestand
 - » Warteraum: Vorbestellliste für nicht sofort erfüllbare Aufträge
 - » Kosten: Lagerhaltungs-, Lieferungs-, Fehlmengenkosten ...
 - Kundenaufträge
 - » Ankunftsverteilung
 - » Artikelbedarf
 - » Wartebereitschaft
 - » Priorität bzw. Präferenzen

Simulation von Lagerhaltungssystemen

• Objekte

- Warensendungen
 - » Auslieferung von Waren, die Kunde beim Lager in Auftrag gegeben hat
- Bestellungen
 - » Zeitliche Verteilung: Bestellzyklus i. d. R. nicht zufällig verteilt
Feste / variable Bestellpunkte
 - » Bestellmenge: Konstant / variabel abhängig von Bestellpolitik
- Lieferungen
 - » Zeitliche Verteilung: Abhängig vom Bestellzeitpunkt und stochastischer Schwankung

Simulation von Lagerhaltungssystemen

• Relationen

- Aufträge => Lieferungen
- Bestellungen => Lieferungen
- Ressourcenfluß:
 - » Sendungen ==> Reduzierung des Lagerbestandes
 - » Lieferungen ==> Erhöhung des Lagerbestandes

Simulation von Lagerhaltungssystemen

- **Aktivitäten**

- Aufgabe von Bestellungen ($\leq$ Bestellpolitik)
 - » (q,t)-Politik:
 - » (S,t)-Politik:
 - » (q,s)-Politik:
 - » (s,S)-Politik:
 - » Lagerbestandsprüfungstermine t für (q,s)-/(s,S)-Politiken

Simulation von Lagerhaltungssystemen

- **Untersuchte Leistungsgrößen**

- Sofortlieferfähigkeit und Kosten abhängig von
 - » Lagerkapazität
 - » Bestellpolitik
- Kosten:
 - » Lagerhaltungskosten
 - » Fehlmengenkosten
 - » Versandkosten
 - » Bestellkosten

Ein Simulationsbeispiel: Mehrprodukt-Lagerhaltungssystem mit Vorbestellungen

• Systemspezifikation

- Lagerhaltungssystem mit Spezialprodukten $i = 1, 2, 3$.
- Exponentialverteilte Zwischenankunftszeiten der Kunden mit Mittelwert 0,1 Monate.
- Nachfragemengen D_i unabhängig, identisch verteilte diskrete Zufallsvariablen.
- Kundennachfrage nach mehreren Produkten gleichzeitig.
- Lieferdauer gleichverteilt zwischen 15 und 30 Tagen.

Ein Simulationsbeispiel: Mehrprodukt-Lagerhaltungssystem mit Vorbestellungen

• Systemspezifikation

Nachfrage- menge D_i	Wahrscheinlichkeiten $P_i (D_i)$		
	Produkt		
	$i = 1$	$i = 2$	$i = 3$
0	0	1/6	1/6
1	1/6	0	1/6
2	1/3	1/6	0
3	1/3	1/3	1/6
4	1/6	0	1/3
5	0	1/6	1/6
6	0	1/6	0

Ein Simulationsbeispiel: Mehrprodukt-Lagerhaltungssystem mit Vorbestellungen

- **Systemspezifikation**

- (s,S) - Bestellpolitik:
- Bei Nachfrageüberhang buchmäßiger Lagerbestand negativ
- Nach Ankunft von Lieferungen erst Befriedigung von offenen Vorbestellungen von Kunden

Ein Simulationsbeispiel: Mehrprodukt-Lagerhaltungssystem mit Vorbestellungen

- **Systemspezifikation**

- Lagerbestände:
- Lagerhaltungskosten:
 - » $LK = h/n \sum L_+(t)dt,$
 - » $FK = j/n \sum L_-(t)dt,$

Ein Simulationsbeispiel: Mehrprodukt-Lagerhaltungssystem mit Vorbestellungen

• Systemspezifikation

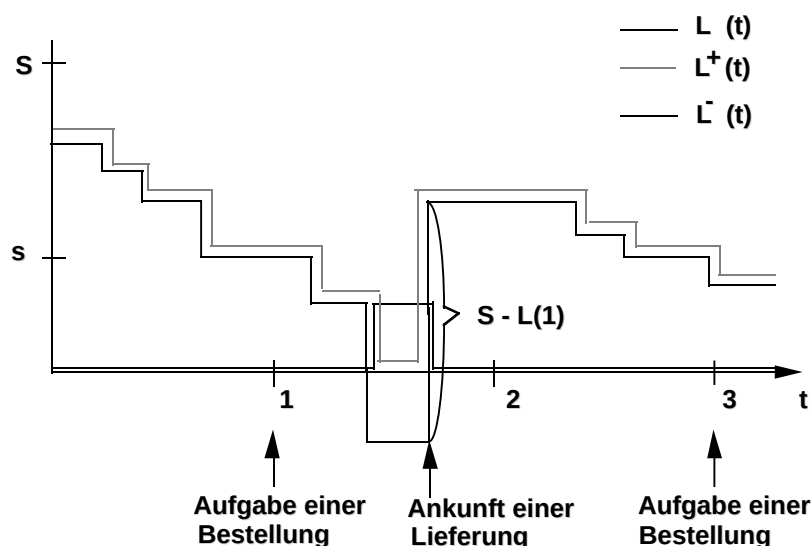
- Anfangslagerbestand $L^+(0) = 60$ je Produktart
- Vorbestellungen $L^-(0) = 0$
- $n = 120$ Monate Simulationszeit
- Analyse von 4 (s, S) - Lagerhaltungspolitiken mit dem Simulationsmodell:

s	20	20	40	60
S	60	100	100	80

Ein Simulationsbeispiel: Mehrprodukt-Lagerhaltungssystem mit Vorbestellungen

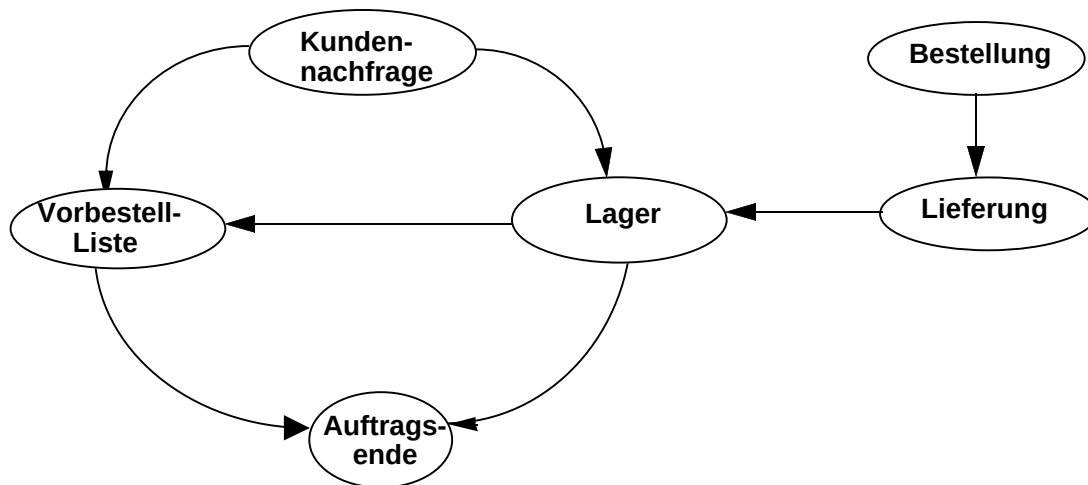
• Systemspezifikation

Beispielhafter Verlauf der $L(t)$, $L^+(t)$ und $L^-(t)$



Modellentwurf des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen

- **Zustandsübergangsdiagramm**



Modellentwurf des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen

- **Prozessorientierter Ansatz umständlich bei reinen Lagerhaltungsmodellen**
 - Objekte vor allem Bestandsgrößen (z.B. Lagerbestand)
 - Keine Modellierung von Aktivitäten (z.B. Bedienungsvorgang), sondern nur von Bestandsänderungen
 - Schwierigkeiten der Identifikation aktiver Objekte (vergleichbar zu Bedienern) in Lagerhaltungssystemen
- **Bestandsänderungen zu Zeitpunkten lassen sich einfacher mit Ereignissen beschreiben**
 - Wahl des ereignisorientierten Ansatzes für Lagerhaltungssystem

Modellentwurf des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen

- Definition folgender Ereignistypen anhand der identifizierten Systemzustandsänderungen
- Ereignis “Lieferung”
- Ereignis “Kundenauftrag”
- Ereignis “Simulationsende”
- Ereignis “Bestellung”
- Priorisierung der Ereignisse in der o.g. Reihenfolge

Modellentwurf des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen

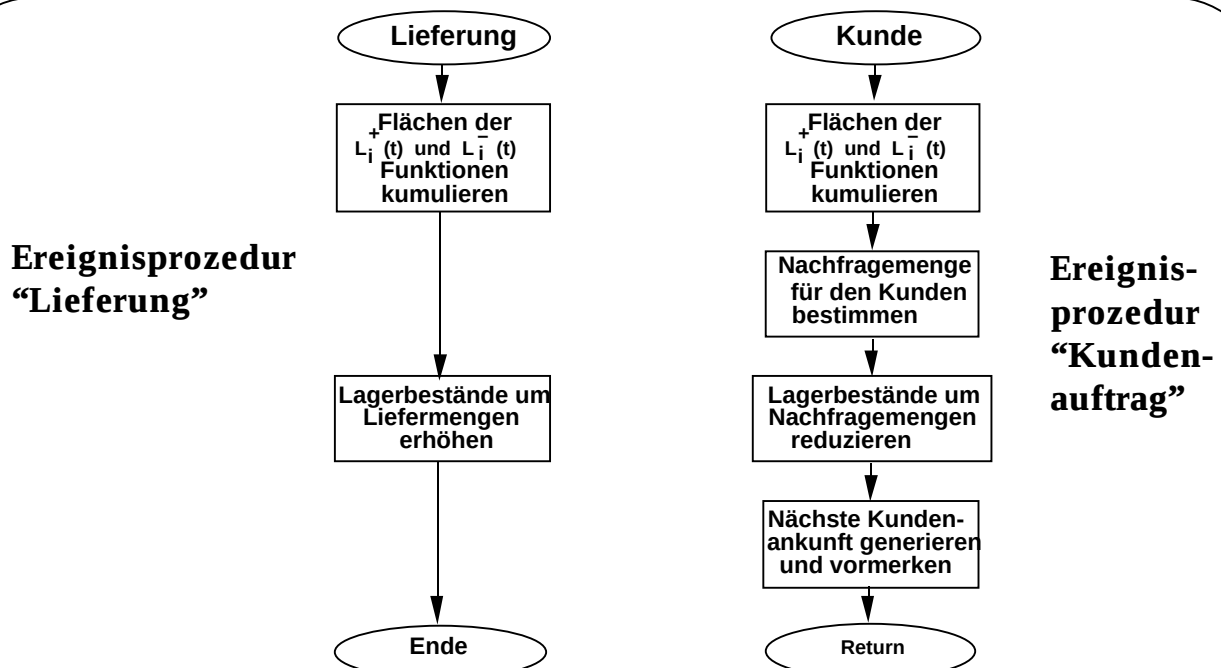
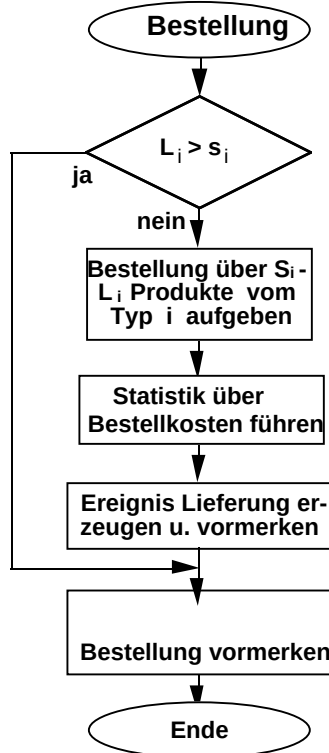


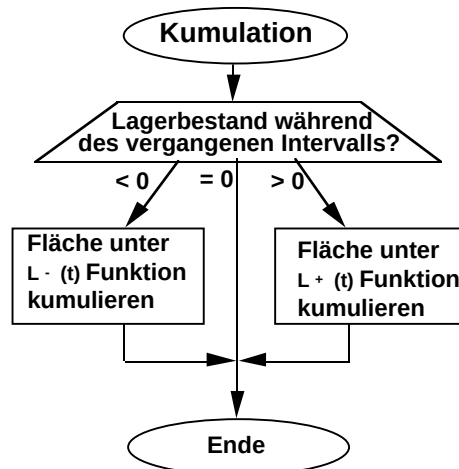
Abb. 2-13: Ereignisroutinen

Modellentwurf des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen



Ereignisprozedur "Bestellung"

Modellentwurf des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen



Hilfsprozedur zur Fortschreibung des Lagerbestandes

Simulationsergebnisse des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen

Simulationsreport des 3-Produkt-Lagerhaltungsmodells

Eingabedaten:

Anzahl Produkte:	3
Anfangslagerbestand je Produkt:	60 Stück
Lagerhaltungspolitik (s,S):	(60, 80)
Simulationsdauer:	120 Monate
Zwischenankunftszeit Nachfrage:	1 Tag
Bestellzeitpunkte:	alle 30 Tage
Lieferfrist (gleichverteilt):	15 bis 30 Tage

Kostenfaktoren der Lagerhaltung (für alle Produkte gleich):

Kosten pro Bestellung:	32 GE
Bestellkosten pro Stück:	3 GE
Lagerhaltungskosten pro Stück und Monat:	1 GE
Fehlmengenkosten pro Stück und Monat:	5 GE

Simulationsergebnisse des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen

Simulationsreport des 3-Produkt-Lagerhaltungsmodells, Forts.

Ausgabedaten:

<u>Kostenart</u>	<u>Produkt 1</u>	<u>Produkt 2</u>	<u>Produkt 3</u>
Bestellkosten (gesamt):	12291 GE	15454 GE	12839 GE
Lagerhaltungskosten (gesamt):	6963 GE	6711 GE	6816 GE
Fehlmengenkosten (gesamt):	2453 GE	6443 GE	3777 GE
Gesamtkosten:	21707 GE	28608 GE	23532 GE
Kosten pro Monat:	181 GE	239 GE	196 GE

Simulationsergebnisse des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen

Tabelle : Simulationsergebnisse für die vier untersuchten Bestellpolitiken

Bestell-politik	Artikel	Gesamt-kosten	Bestell-kosten	Lager-kosten	Fehlmen-gen-kosten
(20, 60)	1	218	100	56	62
	2	277	123	53	101
	3	243	104	60	79
	alle	738	327	169	242
(20, 100)	1	247	116	78	53
	2	322	157	71	94
	3	256	120	81	55
	alle	825	393	230	202

Simulationsergebnisse des Mehrprodukt-Lagerhaltungssystems mit Vorbestellungen

Tabelle : Simulationsergebnisse für die vier untersuchten Bestellpolitiken, Forts.

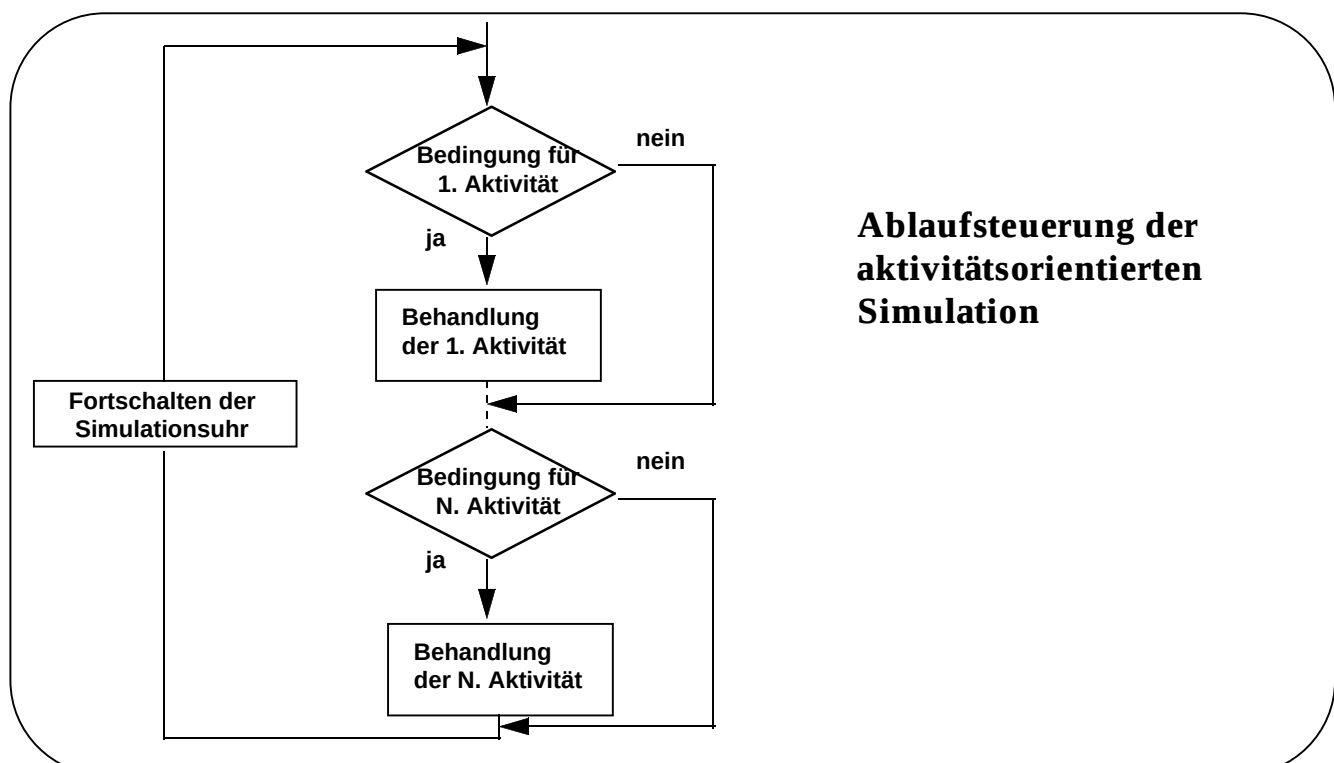
Bestell-politik	Artikel	Gesamt-kosten	Bestell-kosten	Lager-kosten	Fehlmen-gen-kosten
(40, 100)	1	212	95	92	25
	2	283	131	87	65
	3	240	114	89	37
	alle	735	340	268	127
(60, 80)	1	181	102	58	21
	2	239	129	56	54
	3	196	107	58	31
	alle	616	338	172	106

2.9 Andere Modellierungsstile der zeitdiskreten Simulation

• Aktivitätsorientierte Simulation

- Schwerpunkt liegt auf zeitkonsumierenden Handlungen, den **Aktivitäten**
- Aktivitäten einzelner Modellobjekte oder als gemeinsame Handlungen mehrerer Objekte
- Deklarative Formulierung von Testblöcken für Bedingungen
- Fortschalten der Simulationsuhr zu festen Zeitschritten Δt
- Vollständige Überprüfung der komplexen Bedingungen für Beginn/ Ende der Aktivitäten zu jedem Simulationszeitpunkt Δt
- In Abhängigkeit der Prüfergebnisse der Bedingungen Beendigung bzw. Neubeginn zugehöriger Aktivitäten

Aktivitätsorientierte Simulation



Aktivitätsorientierte Simulation

- **Aktivitätsorientierte Simulation**

- In engl. Activity Scanning Simulation
- Ausführung der Aktivität nicht zu vorgemerkttem Zeitpunkt, sondern nach Eintreten beliebig vorgegebener Bedingungen
- Nachteil des aktivitätsorientierten Ansatzes:
- Keine sehr grosse Verbreitung als eigenständiges Weltbild der zeitdiskreten Simulation

Transaktionsorientierter Ansatz

- **Transaktionsorientierter Ansatz**

- Transaktionsorientierung wird teilweise als eigenständiges "Weltbild" der Simulation behandelt, häufig auch als Spezialfall der Prozessorientierung
- Prozessorientierte Simulation:
 - » Gesamte Systemstruktur wird mit allen relevanten Objekten in Form von eigenen Prozessen abgebildet, die jeweils miteinander agieren
 - » Modellierung des Systems durch Betrachtung der Abläufe der einzelnen Objekttypen
- Transaktionsorientierte Simulation:
 - » Modellierung des Systems S aus Sicht eines aktiven Objekttyps, der sich durch S aus anderen (passiven) Objekten bewegt

Transaktionsorientierter Ansatz

- **Transaktionen als temporäre/ dynamische Systemkomponenten**
 - Entsprechend der Prozesse
 - » Lebenszyklus der Transaktionen wird beschrieben
 - » Verwandtschaft zum prozessorientierten Ansatz
 - Suchen sich Weg durch passives Netzwerk von statischen/ permanenten Systemkomponenten, wo sie bestimmten Operationen unterworfen werden
 - Implizite Durchführung von
 - » Warteschlangeneinträgen
 - » Fortsetzung der Transaktionsbewegungen

Transaktionsorientierter Ansatz

- **Transaktionen als temporäre Objekte**
 - Eigenschaften beschrieben durch Parameter
 - » Auf dem Wege durch die Blöcke veränderbar
- **Blöcke als statische Systemkomponenten**
 - Bedienstationen (Facilities),
 - Speicher (Storages) für Lager mit gegebener Kapazität
 - Leisten (Logic Switches) zur Steuerung einzelner Transaktionen

Transaktionsorientierter Ansatz

- **Aufeinandertreffen von Transaktion und Block als Ereignis interpretierbar**
 - Interne Repräsentation als Ereignisse
- **Blockdiagrammtechnik zur Beschreibung des Systemdurchlaufs der Transaktion**
 - Ursprung: Klassische transaktionsorientierte Simulationssprache GPSS
 - Grafische Modellierung auf Basis der Blockdiagramme möglich

Beispiel: Transaktionsorientierte Version des Produktionssystems

- **Repräsentation der Modellkomponenten als Transaktionen und Blöcke**
 - **Fertigungsaufträge = Transaktionen mit Parametern**
 - » Auftrags-Typ
 - » Ankunftszeit
 - » Bearbeitungszeit
 - **Maschinengruppe = Bedienstation**
 - » Ressource mit bestimmter Kapazität
 - » Parameter Bedienungskapazität = Anzahl der Maschinen

Beispiel: Transaktionsorientierte Version des Produktionssystems

- **Grundsätzlicher Ablauf des transaktionsorientierten Modells:**
 - 1) Neuer Auftrag tritt in Produktionssystem ein
 - 2) *Auftrag* reiht sich in die implizite Warteschlange der nächsten *Maschinengruppe* gemäss Bearbeitungsfolge ein
 - 3) *Auftrag* wird bearbeitet

Beispiel: Transaktionsorientierte Version des Produktionssystems

- **Grundsätzlicher Ablauf des transaktionsorientierten Modells, Forts.**
 - 4) *Auftrag* gibt *Maschine* frei
 - 5) *Auftrag* steuert nächste *Maschinengruppe* gemäss Bearbeitungsreihenfolge
 - 6) Auftrag verlässt Produktionssystem

2.9 Bewertung der Modellierungsstile der zeitdiskreten Simulation

- **Ereignisorientierte Sicht:**

- Traditioneller Simulationsansatz
- Typische Systemzustandsänderungen häufig besser mit Ereignissen zu beschreiben (z.B. externe Ereignisse)

- **Vorteile**

- Blick "hinter die Kulissen"
- Einfache Realisierung der Ablaufsteuerung
- Relativ effiziente Simulationsprogramme
- Modellierungsansatz akzeptabel bei einfachen Systemen mit geringen Interaktionen zwischen Modellobjekten

Bewertung der Modellierungsstile der zeitdiskreten Simulation

- **Nachteile der ereignisorientierten Modellierung**

- Verteilung logisch zusammenhängender Abläufe auf verschiedene Ereignisprozeduren
- Ansatz fehleranfällig bei komplexeren Systemen mit größeren Interaktionen zwischen Modellobjekten

- **Vorteile der prozessorientierten Modellierung**

- Direkte und damit natürlichere Repräsentation der Systemobjekte auf Modellebene
- Verwandtschaft zur Objektorientierung
- Grössere Übersichtlichkeit der Systemstruktur

Bewertung der Modellierungsstile der zeitdiskreten Simulation

- **Vorteile der prozessorientierten Modellierung**
 - Höhere Anschaulichkeit insbesondere bei komplexen und überwiegend parallelen Handlungen
 - Besonders gut geeignet für typische Warte-/ Bedienungssysteme
- **Nachteile**
 - Teilweise umständliche Modellierung bei anderen Modelltypen

Bewertung der Modellierungsstile der zeitdiskreten Simulation

- **Aktivitätsorientierte Sicht**
 - Wenig verbreitet als eigenständiger Ansatz
 - Effizienzprobleme
- **Transaktionsorientierte Sicht**
 - Spezialfall der Prozessorientierung
 - Sehr kompakte Modellierung
 - Eingeschränkte Anwendbarkeit auf sehr einfache, spezielle Bedienungssysteme:
 - Flexibilität der Modellierung eingeschränkt

Bewertung der Modellierungsstile der zeitdiskreten Simulation

• Schlussfolgerungen

- Prozessorientierte Simulationsmodellierung vorzuziehen !
- Andere Modellklassen ggf. mit ereignisorientiertem Ansatz
- Ereignisorientierter Ansatz grundlegend für Verständnis der internen Simulationsabläufe !
- Kopplung jeweils geeigneter Modellierungsstile für Teilaspekte eines Modells wünschenswert

2.10 Objektorientierte Simulation

- **Objektorientierte Simulation allgemein als**
 - Simulationsmodellierung eines Systems mit objektorientierten Entwurfs- und Implementierungsmethoden und -werkzeugen
- **Objektorientierter Ansatz**
 - Darstellung der Entitäten der realen Welt nicht allein durch Daten, sondern auch durch Operationen, die auf diesen Daten arbeiten
 - Ergebnis dieser Kombination = **Objekt**
 - Ganze Gruppe gleichartiger Objekte = **Objektklasse**
- **Objekte im Sinne OO- Programmiersprachen**
 - Erste OO-Programmiersprache *Simula* im Bereich der Simulation
 - Weitere Beispiele: (C++), Smalltalk, Eiffel...

Grundlagen der Objektorientierten Simulation

- **Objekte sind**
 - Einkapselung von
 - » Strukturinformation, Datenstrukturen, Attributen
 - » Verhaltensregeln, prozedurale Beschreibungen von Verhalten
 - Eindeutig identifizierbar
 - Klar abgrenzbar gegenüber ihrer Umgebung
 - In Kommunikation mit anderen Objekten mittels **Nachrichtenaustausch**
- **OO-Software als Sammlung einzelner Objekte, die sowohl **Daten- als auch Verhaltensbeschreibung** enthalten**

Grundlagen der Objektorientierten Simulation

- **Objektklassen**
 - **Vererbung** gemeinsamer Eigenschaften über **Klassenhierarchie**
 - **Modifizierbarkeit und Erweiterbarkeit** von Objekten durch **Klassenkonzept** unterstützt
 - » **Modifikation** der Originalklasse durch separate Ergänzung
 - » **Subklasse** mit Voreinstellung aller Eigenschaften der Oberklasse und mit zusätzlichen Attributen bzw. Methoden
 - » **Modifizierung** durch Erweiterungen ohne Rückwirkung auf ursprüngliche (Ober-) Klasse
 - » **Wiederverwendbarkeit** bereits vorhandener Programmkomponenten

Grundlagen der Objektorientierten Simulation

- **Objekte und Objektklassen**
 - Die reale Welt lässt sich präziser beschreiben als allein mit Daten
- **Objektorientierung in der Simulation**
 - OOP in der Simulationssoftwareentwicklung
 - Objektorientierte Modellierungsmethoden

Grundlagen der Objektorientierten Simulation

- **Abgrenzung zur konventionellen Simulation**
 - Konventionell: Nutzung vordefinierter Sprachkonstrukte, die nicht veränderbar oder erweiterbar sind
 - Keine Möglichkeit der Definition, Verknüpfung, Erweiterung oder Wiederverwendung dieser Konstrukte
- **Bezug der Objektorientierung zur prozessorientierten Sicht der Simulation**
 - Prozessorientierung der Simulation als Sonderfall der Objektorientierung, sofern
 - » nur Prozessobjekte
 - » Interprozesskommunikation ausschliesslich über Nachrichtenaustausch

Wie funktioniert Objektorientierte Simulation ?

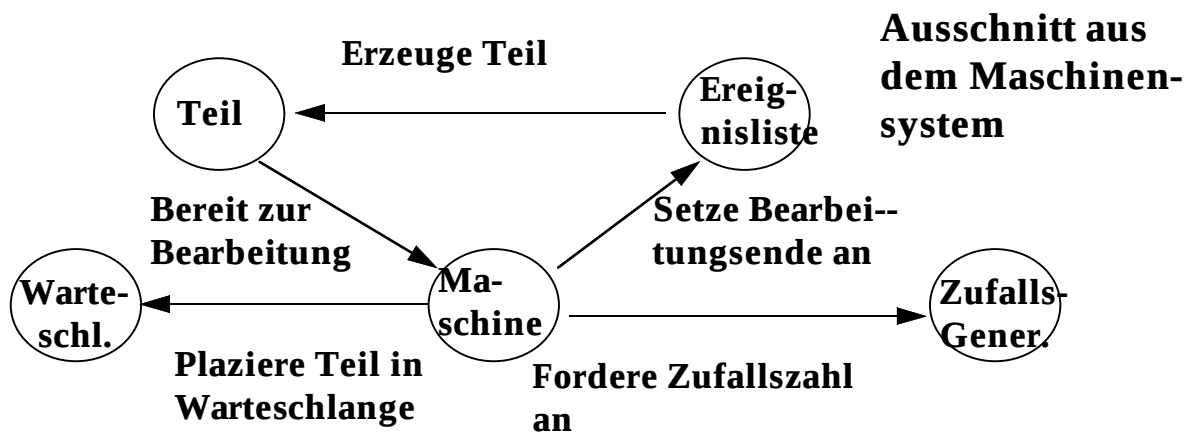
- **Beispiel: Einfaches Maschinen-Bedienungssystem**
- **Objektorientierter Ansatz für das Maschinenbeispiel**
 - Kommunikation über Nachrichtenaustausch von Objekten, die in Klassen zusammengefasst sind
 - Definition folgender Klassen:
 - » Ereignisliste
 - » Zufallszahlengenerator
 - » Fertigungsteil
 - » Maschine
 - » Warteschlange

Wie funktioniert Objektorientierte Simulation ?

- **Nachrichtenaustausch der Objekte und Ausführung von Methoden im Maschinenbeispiel**
 - Nachricht des *Ereignislisten*-Objektes an Objekt *Teil*, von sich mit Erzeugungsmethode einen Nachfolger zu generieren
 - Meldung an die *Maschine*, dass *Teil* bereit zur Bearbeitung ist
 - Methode des Objektes *Maschine* prüft mit entsprechender Methode, ob sie Bearbeitungskapazität zur Verfügung hat

Wie funktioniert Objektorientierte Simulation ?

- **Abbildung zur Kommunikation zwischen den Objekten im Maschinenbeispiel**



Vorteile bzw. Potential der Objektorientierung für die Simulation

- **Konzeptuelle Nähe zum Problembereich**
 - Nahe, natürliche Abbildung von realen Objekten auf programmiersprachliche Objekte
 - Einfachere Nachvollziehbarkeit des Zusammenhangs zwischen Programmstruktur und Struktur des konzeptuellen Modells bzw. Realsystems
 - Vereinfachung der Modellvalidierung durch grössere Anschaulichkeit
 - Einheitliches Konzept für Analyse, Design und Programmierung bei Verwendung objektorientierter Entwurfsmethoden

Vorteile bzw. Potential der Objektorientierung für die Simulation

- **Sichtweise der Simulation als Sendung und Empfang von Nachrichten zwischen Objekten**
- **Wiederverwendbarkeit von Objekten**
 - Modellbildung als "Komposition von Modellbausteinen"
 - Vereinfachte Verfeinerung / Modifizierbarkeit existierender Modelle durch Realisierung von Objekten als Klassen und Nutzung des **Vererbungsmechanismus**
 - Basis für den Aufbau von Modellsammlungen
- **Verbesserung der Integrationsfähigkeit von Modellsammlungen**
 - Vereinheitliche Kommunikation
 - Erweiterbares, integrierbares Konzept der Klassenhierarchie

Vorteile bzw. Potential der Objektorientierung für die Simulation

- **Erleichterung der Entwicklung von Simulatoren**
 - Durchgängiges Konzept bei Entwurf und Implementierung
 - Geringere Kosten bei Änderung
 - Bessere Basis für interdisziplinäre Arbeit
- **Unterstützung der hierarchischen Modellierung**
 - durch Zerlegung in Subsysteme
 - Schrittweiser Aufbau einer Klassenhierarchie
- **Vereinfachung der Verteilung bzw. Parallelisierung von Modellen**
 - Konzept aktiver Objekte entspricht Verteilung der Prozesse
 - Kapselung und Nachrichtenaustausch setzt gemeinsamen Speicher nicht voraus

Probleme und Grenzen der Objektorientierten Simulation

- **Sicherheitsrisiken für Simulationsprogramme**
- **Abbildung kontinuierlicher Kommunikation nur schwer möglich**
- **Teilweise Laufzeitprobleme**
- **Zusatzaufwand durch Umstellung von ausgereifter konventioneller Simulationstechnik und -software**

Probleme und Grenzen der Objektorientierten Simulation

- **Bisher geringe Erfahrungen mit der Anwendung objektorientierter Modellierungsmethoden**
 - Simulationsmodellierung, nicht Entwurf von Simulationssoftware
 - z.B. Unified Modelling Language (UML)
 - Unified Modelling Language UML:
- **Fehlende Entwurfsmethoden/ Standards**
 - Problem: Grosse Freiräume der Entwickler bei der OO-Implementierung von Simulationsklassen

Bewertung der Objektorientierten Simulation

- **Fazit**

- Kombination von Klassenkonzept und Nachrichtenaustausch
- Flexibilität
- Wiederverwendbare Modellbausteine
- Nützlich bei umfangreichen, komplexen, wiederverwendbaren Modellen

» **Literatur: D. Hill: Object-Oriented Analysis and Simulation. Addison-Wesley, Cambridge 1996**